Decision support for management of parallel database systems

Citation for published version:
Williams, H, Zhou, S, Taylor, H & Tomov, N 1996, Decision support for management of parallel database systems. in *Proceedings of High Performance Computing and Networking.* vol. LNCS 1067, Lecture Notes in Computer Science, Springer, pp. 164-169. https://doi.org/10.1007/3-540-61142-8_543

Digital Object Identifier (DOI):
10.1007/3-540-61142-8_543

Link:
Link to publication record in Heriot-Watt Research Portal

Document Version:
Early version, also known as pre-print

Published In:
Proceedings of High Performance Computing and Networking

Title: **Decision Support for Management of Parallel Database Systems**

Author: **M. Howard Williams, Shaoyu Zhou**
Hamish Taylor and **Neven Tomov**
Presenter: to be decided by 23/02/96
Contact Address: Department of Computing and Electrical Engineering
Heriot-Watt University, Riccarton, Edinburgh,
EH14 4AS, U.K.
E-mail: howard@cee.hw.ac.uk zhou@cee.hw.ac.uk
Tel: +44 131 451 3430
Fax: +44 131 451 3431

Decision Support for Management of Parallel Database Systems

M. Howard Williams, Shaoyu Zhou, Hamish Taylor and Neven Tomov

Department of Computing and Electrical Engineering, Heriot-Watt University, Riccarton, Edinburgh, EH14 4AS, U.K.

Abstract. Parallel database systems are generally recognised as one of the most important application areas for commercial parallel systems. However, the task of managing the performance of a parallel database system is exceedingly complex. The initial choice of hardware configuration to support a particular DBMS application and the subsequent task of tuning the DBMS to improve performance rely not only on the way in which the data is structured, but also on how it is fragmented, replicated and distributed across the processing elements of the system. To understand the behaviour of a particular application requires the study of large volumes of performance data. To simplify this process it is essential to provide some means of presenting performance data in a comprehensible form which will aid visualisation. This paper explores some of the issues relating to decision support for the performance management of parallel database systems and describes an analytical capacity planning tool to assist users in this task.

1 Introduction

Database systems provide an obvious application area for parallel computer systems. In the next few years commercial information processing is expected to achieve its increasing requirements for high performance by moving database systems from mainframe computers to parallel machines.

However, even for database systems running on conventional machines, performance management is a non-trivial task. System administrators need to make the right decisions when tuning the system to achieve better performance for particular applications, or when introducing new services or changes to an existing installation to achieve efficient performance.

When database systems are moved to parallel environments, the process of setting them up and tuning them to obtain better performance is made more complex still. Not only does one have to make decisions on how to structure data, but also how to fragment data and tasks and distribute them across different processing elements (and even across different devices associated with these processing elements). A recent study [8] showed how different data placement strategies can have a major impact on performance, even in the case of a simple benchmark such as TPC-B. In the case of more complex workloads, it may be very difficult to determine the cause of any loss of performance, and to discover to what extent performance can be improved.

Apart from data structuring and layout, it is essential to provide simple yet flexible ways of presenting the performance information to enable the systems personnel of parallel database systems to visualise the consequences of decisions. Large quantities of performance data accumulate during the performance monitoring process for systems management personnel to analyse. Their task, which involves understanding the data and drawing conclusions from it, is non-trivial. Of particular importance for determining overall performance is the identification of the bottleneck resources under a particular workload, and how they are affected by changes to the system.

This paper explores some of the issues relating to decision support for the performance management of parallel database systems. It focuses on the aspects of data placement effectiveness and visualisation of performance data, and describes an analytical performance estimator for parallel database systems which has been developed to assist capacity planning of parallel database systems. The development of this analytical tool has tackled the above mentioned issues, through a mechanism for users to manipulate different data placement strategies in order to assess their relative merits and a graphical performance visualiser for analysing the large volume of estimated performance data.

2 Data Placement Effectiveness

Data placement has a significant impact on the overall system performance in parallel database systems. Moreover, the task of deciding how to fragment and distribute the data of a large database is complex. To aid the user various data placement strategies have been developed by researchers which will enable this task to be performed mechanically [1, 2, 3]. However, there is no simple way of determining which strategy would provide the best results for any particular database. As shown in [8], a data placement strategy performs differently under different circumstances. For transaction benchmarks such as TPC-B and TPC-C, relatively small changes in configuration can have a significant impact on the performance obtained from the data distribution generated by a particular data placement strategy. This task becomes even harder when real applications are considered, which are more complex than these simple benchmarks.

One factor which may be important in deciding on the best data placement strategy is the sensitivity of the system to changes in the workload. Small changes in the relative frequencies of queries may have significant consequences for the performance of the overall system. Equally changes to the data stored caused by updates to the database may result in a skewed data distribution and hence an unbalanced load on processing elements which in turn degrades the overall performance of the system. Another factor which affects the choice of data placement strategy is database size. A study [6] conducted on the sensitivity of data placement strategies to changes in database size demonstrates clearly that such changes may have different effects on different types of data placement strategies. Some strategies are more sensitive than others. Moreover, sometimes a decrease in database size does not necessarily improve performance when certain types of data placement strategies are used. The task of determining an efficient strategy which is least susceptible to changes is exceedingly complex.

In practice, it is clearly impossible for a system administrator to perform experiments with different data placement strategies on a real data set. The scale of the task of rearranging the data for different experiments is in itself prohibitive, let alone the subsequent measurements and analysis of performance. Thus, if the system administrator is to assess the effectiveness of different data placement strategies (or even of different parameters for the same strategy) for a particular application running on a particular hardware configuration, this can best be achieved by the use of some form of performance prediction tool.

The sizing of a new application and the migration of an existing application from a conventional database to a parallel platform are especially in need of assistance in the form of such a tool, so that a cost effective parallel DBMS configuration and an efficient data layout can be obtained prior to the sizing or migration.

3 Visualisation of Performance Data

A major problem facing systems personnel is that of visualising what is happening to data in the process of data placement, and the effect which this has on creating bottlenecks within the system for a particular workload. Systems personnel generally have access to large volumes of performance data produced as part of the monitoring process. However, making sense of such data and drawing relevant conclusions from them is a non-trivial task. Determining the bottlenecks in the system for a particular workload and the effect different changes to the system will have on these bottlenecks is an important step in determining the overall performance.

It is essential to provide simple yet flexible ways of presenting the performance information to enable the systems personnel to visualise the consequences of decisions. Graphical representation of resource utilisation for a particular configuration and data layout can be used to see rapidly where bottlenecks arise and whether loads can be re-distributed to avoid them. Comparisons between the utilisations arising from different configurations, different data layouts and even changes to the workload can be represented graphically in such a way that the user can understand quickly and easily the effects of different decisions.

Suitable ways of visualising performance data can help to improve understanding of the data and make the analysis easier. With visual displays, there are opportunities for showing relationships by proximity, by containment, by connected lines, or by colour coding [5]. However, effective visualisation requires considerable knowledge of the key measures and their relationships if it is to help user understanding and analysis of the data.

In developing interfaces to provide effective visualisation, account must be taken of the needs of different users. In particular, two classes of users have an interest in using such a tool for different purposes. On the one hand, sales personnel need to estimate the performance of a client's application on different hardware configurations in order to determine an appropriate configuration for tender (application sizing). On the other hand, a systems administrator needs to tune the performance of a particular application running on a particular machine configuration. While both types of users need to be able to experiment with different data placement strategies, the nature of the experiment and the way in which results are presented will differ. For the salesperson, detailed understanding of the performance of the application is not essential, and the main objective is to experiment with different data placement strategies across different machine configurations. For the system administrator details such as the utilisation of individual resources and throughput and a clear picture of how a bottleneck arises in the system are of greater importance.

4 Performance Prediction

A number of techniques can be used to predict performance using measures such as throughput, response time and utilisation to handle a particular workload on parallel database architectures. Analytical modelling and simulation are the two most promising approaches. An analytical tool provides predictions for average or typical behaviour. It is easier to use and quicker at estimating low-level DBMS and hardware component performance. Simulation methods are more accurate in modelling the dynamic nature of parallel DBMSs, such as contentions, but this is usually at the cost of time-consuming simulation runs which give specific rather than general behaviour.

A set of tools has been developed, referred to as STEADY (System Throughput Estimator for Advanced Database sYstems), based on analytical modelling [7] to aid a user in selecting a data placement strategy, in determining the effects which changes to the system configuration might have on the performance for a particular workload and in displaying performance information such as the system throughput, bottlenecks, cost distribution and resource utilisation in ways which are easy for a system manager to understand and use.

The approach adopted separates the performance model into 3 parts:

1. The application layer includes models for the specific database relations used in the application, the database queries for the application, and is concerned with the way in which the DBMS is used and the way in which the data is distributed.
2. The DBMS kernel layer contains cost models for basic data operators implemented in the DBMS kernel, including models for logging and locking requests.
3. The platform layer consists of models of low-level platform components such as the file system, lock manager, operating system and hardware.

Of these the application layer is relevant to the systems personnel responsible for the management of a particular database application.

STEADY provides a cost-effective means to cope with experimentation among different data placement strategies for an application running on a parallel database system. It supports a range of data placement strategies to choose from.

STEADY has been set up to be used in two different modes:

1. In stand-alone mode it can be used to investigate different data placement strategies, different DBMS configurations, different queries or even different databases by setting up the parameters for each experiment through the normal interface and running the experiment manually. This mode allows the user to investigate the effects of a particular set of input parameter values in detail. The user specifies the set of parameter values for each experiment and then, depending on the results, adjusts the values to obtain a better performance. This is particularly useful in performance tuning, in which small changes to DBMS configuration, data layout or queries are usually expected. It is also useful in performance studies in which the search space of input parameter values cannot be easily defined.
2. Coupled with the search tool Testpilot [4] STEADY can be used to explore some search space automatically. This mode enables the users to conduct comparative studies among different sets of input parameter values. Users can specify the search space of input parameter values as well as the performance measurements (e.g. throughput) through the session manager of STEADY. Testpilot will then select experimentation points from the search space and conduct the experiments by invoking relevant modules of STEADY automatically. The resulting performance data is stored in such a way that various comparative studies can be performed on the response values of the performance measurements for different experimentation points in the search space. In this way, the tedious task of executing experiments repeatedly through the normal interface of STEADY has been taken over by Testpilot and users can concentrate more on the interpretation of the results.

The multi-level graphical user interface provided by STEADY can be tailored to suit the needs of different types of users. Users can choose which performance data, such as throughput, bottleneck and resource utilisation, they are interested in and conduct various comparisons among them. They can view the results comparatively through a display which shows the relative performance of different sessions or through a clickable hardware configuration graph, which highlights the bottlenecks in the system, illustrates utilisation of each resource and allows users to fine tune data placement through direct manipulation of data fragments represented on the screen. Moreover, users can also trace a multi-level cost tree to investigate the distribution of costs among different functional components or even software processes, such as I/O, data operations, locking and logging.

5 Conclusions

The understanding and management of the performance of parallel database systems is an area of work still in its infancy. There are few tools available to assist users or service suppliers to make the right decisions for performance management of parallel database systems.

An analytical performance estimation tool provides a cost-effective means to cope with the demand of performance management of parallel database systems and has the potential to be exploited not only by pre-sales support and service delivery personnel of a parallel DBMS vendor, but also by the customers who need to size new applications or migrate existing applications from conventional DBMS to a parallel platform.

A visualisation component of such an analytical tool helps both the sales and system personnel to interpret large volumes of performance data and identify the potential problems affecting the performance before they occur. Moreover, in today's rapidly growing market of parallel database systems, this is particularly useful in providing an effective experimental environment to understand the behaviour of such systems by self study and through formal training.

Acknowledgements

The authors acknowledge support from the Commission of the European Union for the work reported in this paper which reflects work done under the Pythagoras project (ESPRIT III EP7091) and the Mercury project (ESPRIT IV 20089). They also wish to thank Mr Phil Broughton, Mr Arthur Fitzjohn and Mr John Hayley of ICL, Prof Martin Kersten of CWI (Amsterdam) and Mr Albert Burger of Heriot-Watt University for their assistance and support.

References

1. G. Copeland, W. Alexander, E. Boughter, and T. Keller. Data placement in Bubba. In *Proceedings of ACM SIGMOD Conference*, 1988.
2. K. Hua, C. Lee, and H. Young. An efficient load balancing strategy for shared-nothing database systems. In *Proceedings of DEXA'92 conference*, pages 469–474, Valencia, Spain, September 1992.
3. M. B. Ibiza-Espiga and M. H. Williams. Data placement strategy for a parallel database system. In A. M. Tjoa, I. Ramos, Springer-Verlag, and Vienna, editors, *Proceedings of Database and Expert Systems Applications 92*, pages 48–54, Spain, 1992.
4. M. Kersten and F. Kwakkel. Design and implementation of a DBMS performance assessment tool. In *Proceedings DEXA'93*, pages 265–276, Praag, September 1993.
5. B. Schneiderman. *Designing the User Interface - Strategies for Effective Human-Computer Interaction.* Addison-Wesley Publishing Co., 1992.
6. S. Zhou and M. H. Williams. Data placement in parallel database systems. In M. Abdelguerfi, Y. Kambayashi, and K.-F. Wong, editors, *To appear in Parallel Database Techniques.* IEEE Press, 1996.
7. S. Zhou, M. H. Williams, and H. Taylor. Practical throughput estimation for parallel databases. *submitted for publication to Software Engineering Journal*, 1994.
8. S. Zhou, M. H. Williams, and H. Taylor. A comparative study of data placement in a parallel DBMS. *submitted for publication to IEEE Transactions on Computers*, 1995.